

### Discussion Topics

Ir1 provides an event logging capability at each site.

#### OBJECTIVES:

This lesson will:

- introduce different components of event logs
- provide troubleshooting tips using the logs
- describe operational details

At the end of this lesson, you will be

- familiar with various interfaces for logging
- able to read records in a log file
- provide a better description of a problem in a NCR

## Overview



- Ir1 applications record events (success/failure) to log files
- Interface is provided by the Event Logger
- Option is available for Openview notification
- Events are logged to an ascii (text) file
- Event data is periodically transferred to a database for archival
- The ascii log file may be browsed using any available unix tools (grep, awk, etc.)
- Database browsing with sql

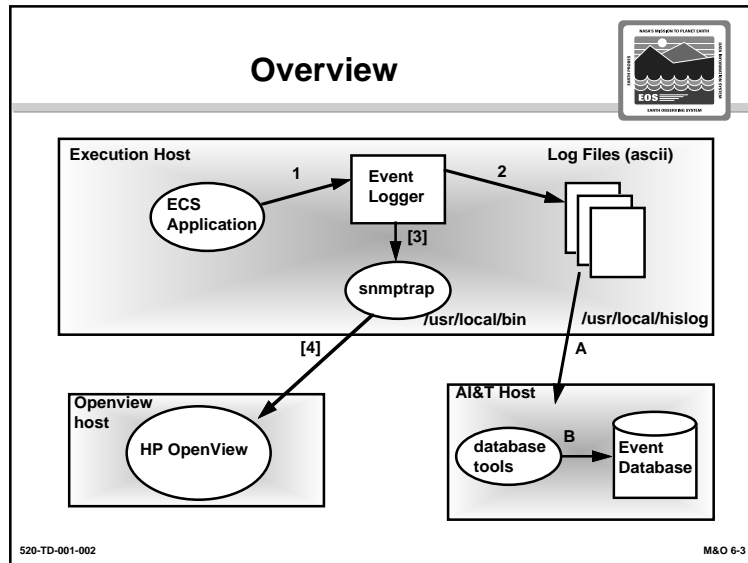
520-TD-001-002

M&amp;O 6-2

## Discussion Topics

Event Log overview -- used to collect data and report problems

- Ir1 applications record events (success/failure) to log files
- Interface is provided by the Event Logger
- Option is available for Openview notification
- Events are logged to an ascii (text) file
- Event data is periodically transferred to a database for archival
- The ascii log file may be browsed using any available unix tools (grep, awk etc.)
- Database browsing with sql



### Discussion Topics

**Step 1** - Application calls event logger

**Step 2** - Logger records event in the log file (only one)


**Step 3 (optional)** - If the disposition dictates (e.g. fatal error), the event is sent to snmptrap

**Step 4 (optional)**- snmptrap sends event to Openview.

**Step A** - Log files are transferred (ftp'd) to the AI&T host (Sybase server)

**Step B** - Files are loaded to database

## Log File



- Each application has its own event log file with an extension .log (e.g., IngestLocal.log)
- Files are located in path /usr/local/hislog
- Data is recorded in ascii (text)
- Each file contains several records
- Each record has multiple fields separated by a delimiter (vertical bar '|')
- Every record is terminated by a newline (\n)
- Log files should not be modified
- Copy file and edit the copy if needed
- Writing to the file is synchronized using UNIX file locking

520-TD-001-002M&O 6-4

---

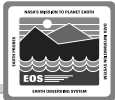
### Discussion Topics

---

Event Log characteristics:

- Each application has its own event log file with an extension .log (e.g., IngestLocal.log)
- Files are located in path /usr/local/hislog
- Data is recorded in ascii (text)
- Each file contains several records
- Each record has multiple fields separated by a delimiter (vertical bar '|')
- Every record is terminated by a newline (\n)
- Log files should not be modified
- Copy file and edit the copy if needed
- Writing to the file is synchronized using UNIX file locking

## Log File: Record Format



**num|sev|disp|time|an|av|pid|osn|osv|hn|ip|uniqid|Msg|&n**

<b>num*</b>	= event number (application specific)
<b>sev*</b>	= event severity
<b>disp*</b>	= event disposition
<b>time&amp;</b>	= event timestamp
<b>an*</b>	= application name (string)
<b>av*</b>	= application version (string)
<b>pid</b>	= process id
<b>osn</b>	= operating system name
<b>osv</b>	= operating system version
<b>hn/ip</b>	= name and ip address of the execution machine
<b>uniqid</b>	= id for making the record unique in the database
<b>Msg&amp;</b>	= application-provided text message

Key: \* = provided by application, & = optionally provided by application

520-TD-001-002
M&O 6-5

## Discussion Topics

### Event number

- Defined by the applications (4 bytes unsigned)
- Not very useful at this time since it is not unique

### Event Severity

- Categorizes an event from success to fatal error
- Values are:
  - 0=success, 1=warning, 2=error, 3=fatal error, 4=message, 5=user information, 6=notice

### Event Disposition

- Defines how the event should be handled by the logger
- Openview notifications are based on this field
- Values are:
  - 0=local log only, 1=openview critical, 2=openview normal

### Event Timestamp - when the event occurred

### Application name/version

- Primarily for tracking events for multiple applications writing to a single log file (not Ir1)
- Some Ir1 applications use this for logging from different threads (Gateway.log)

### OS Name/Version and Host Name/IP Address

- Constants for all records in a given log file
- Useful when data is consolidated in the database

### Process Id

- A unique identifier for an application
- Useful for browsing the log
- Can be used to get events logged by a specific instance of an application


### Unique Id

- Number making a record unique in the database (two identical records may be logged in a second)
- Starts from process id and incremented every time an event is logged

### Message

- Optional message provided by the application
- Gives a textual description of the event (use this instead of event number)
- Displayed by Openview in a popup window for critical events (disposition 1)

## Sample Log File (Gateway.log)



```
2260[3]1|12/22/95 15:39:54|GWProxyMain|1.0|100|SunOS|5.4|kingkong|155.157.99.26|100|
Gateway Proxy Exception: Unexpected DAA message received|
2225[0]0|12/22/95 15:43:42|Gateway|1.0|29977|SunOS|5.4|kingkong|155.157.99.26|29984|Proxy 111 started on socket 5|
2260[3]1|12/22/95 15:44:23|GWProxyMain|1.0|111|SunOS|5.4|kingkong|155.157.99.26|111|
Socket Exception: Communication failure (connection broken with peer) : errno 22|
2225[0]0|12/22/95 15:44:53|Gateway|1.0|29977|SunOS|5.4|kingkong|155.157.99.26|29985|Proxy 114 started on socket 5|
2260[3]1|12/22/95 15:44:55|GWProxyMain|1.0|114|SunOS|5.4|kingkong|155.157.99.26|114|
Gateway Proxy Exception: Unexpected DR message received|
2225[0]0|12/22/95 15:45:36|Gateway|1.0|29977|SunOS|5.4|kingkong|155.157.99.26|29986|Proxy 117 started on socket 5|
2222[0]0|12/22/95 15:45:42|GWProxyMain|1.0|117|SunOS|5.4|kingkong|155.157.99.26|117|
Authentication succeeded for user nvazarka|
1000[5]0|12/22/95 15:45:43|GWProxyServer|1.0|117|SunOS|5.4|kingkong|155.157.99.26|117|Gateway proxy server started|
2225[0]0|12/22/95 15:50:19|Gateway|1.0|29977|SunOS|5.4|kingkong|155.157.99.26|29987|Proxy 121 started on socket 5|
```


520-TD-001-002M&O 6-6

## Discussion Topics

### Note:

- Original log modified to fit in the chart (new lines added)
- Two threads (GWProxyMain, GWProxyServer) and Two processes (Gateway and Gateway Proxy) are logging to the same file.
- The unique id for process 29977 (Gateway). It is incremented for each record
- OS name/version and IP address are same for all records
- Application name and Version differentiate between different threads and processes
- Message provides a description of the event
- Disposition tells us if the event was sent to Openview

### Openview Interface



- Alerts operator of a critical condition in order to take corrective actions
- Notifications sent to Openview using custom program snmptrap (in /usr/local/bin on every host)
- All Openview notifications seen in the ECS Application Event Log
- Applications decide disposition
- Two kinds of notifications:
  - disposition 1 (critical, high priority)
  - disposition 2 (normal, still requires operator intervention)

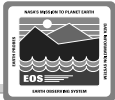
520-TD-001-002 M&O 6-7

## Discussion Topics

### HP OpenView:

- Alerts operator of a critical condition in order to take corrective actions
- Notifications sent to Openview using custom program snmptrap (in /usr/local/bin on every host)
- All Openview notifications seen in the ECS Application Event Log
- Applications decide disposition
- Two kinds of notifications:
  - disposition 1 (critical, high priority)
    - » Appears as a critical message (bold) in the openview log and a popup window with the custom message is displayed
  - disposition 2 (normal, still requires operator intervention)
    - » Only message to the openview log

## Database Interface



- Event data is inserted into database located at the AIT server at every DAAC
- Only \*.log files in /usr/local/hislog are exported
- Database contains approximately a month of data
- Cron jobs automate the database import process
- Two types of cron scripts
  - On every host for transferring files to AIT Server
  - On AIT Server to insert the ascii files into database
- Canned reports are available
- Custom reports may be created using sql

520-TD-001-002M&O 6-8

---


### Discussion Topics

**Database Interface** -- the most important of the logging system

- Event data is inserted into database located at the AIT server at every DAAC
- Only \*.log files in /usr/local/hislog are exported
- Database contains approximately a month of data to look at
- Cron jobs automate the database import process
- Two types of cron scripts -- every IR1 host will run a script that transfers and loads ascii files
  - On every host for transferring files to AIT Server
  - On AIT Server to insert the ascii files into database
- Canned reports are available
- Custom reports may be created using sql command



## Database Interface (cont'd)



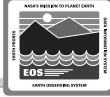
- Files are located in /lr1\_IT/CSS/bin
- Cronjob on every lr1 machine (CSScron.transfer)
  - Runs under username logftp (transfer.csh script)
  - Set path to /usr/local/hislog
  - Move all log files (\*.log) to ./download with a unique stamp (custom stamp program)
    - » Example: Gateway.log is copied as 'hostname.mmdd.pid.Gateway.log'
  - Copy (ftp) all files in ./download to /usr/local/hislog/upload on AIT host (custom transfer program)
  - Delete the copied files
  - Log kept in file ./download/hostlog
    - » Check for errors periodically

520-TD-001-002M&O 6-9

### Discussion Topics

- Cronjob on every lr1 machine (CSScron.transfer)
  - Runs under username logftp (transfer.csh script)
  - Set path to /usr/local/hislog
  - Move all log files (\*.log) to ./download with a unique stamp (custom stamp program)
    - » Example: Gateway.log is copied as 'hostname.mmdd.pid.Gateway.log'
  - Copy (ftp) all files in ./download to /usr/local/hislog/upload on AIT host (custom transfer program)
  - Delete the copied files
  - Log kept in file ./download/hostlog
    - » This log should be checked for errors approximately once per week

## Database Interface (cont'd)



- **Cronjob on AIT host (CSScron.dbload)**
  - Runs under username logdbusr (dbload.csh script)
  - Set path to /usr/local/hislog/upload (ftp'd files)
  - Load all log files (\*log) to database using 'bcp'
  - Remove (delete) files that were successfully transferred to database
  - Log kept in file ./hostlog
    - » Check for errors periodically
- **Database information**
  - Name is 'eventlogdb'
  - Table name is 'eventlog'

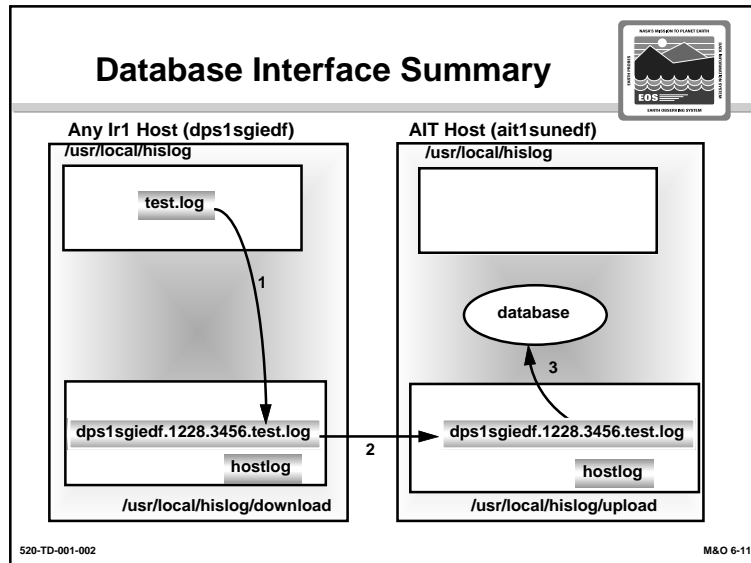
520-TD-001-002M&O 6-10

---

### Discussion Topics

---

- Cronjob on AIT host (Sybase) (CSScron.dbload)
  - Runs under username logdbusr (dbload.csh script)
  - Set path to /usr/local/hislog/upload (ftp'd files) - where all log files go
  - Load all log files (\*log) to database using 'bcp' (bulk copy command)
  - Remove (delete) files that were successfully transferred to database
  - Log kept in file ./hostlog
    - » This log should be checked for errors periodically.
- Database information
  - Name is 'eventlogdb'
  - Table name is eventlog
  - You will need a password for database access



### Discussion Topics


**Step 1:** The test log file is copied to the `/usr/local/hislog/download/` directory.

**Step 2:** The test log file is copied to the upload directory (`/usr/local/hislog/upload`)

Note: Steps 1&2 occur by using the same cron job "CSScron.transfer"

**Step 3:** The file data is then inserted into the database using the cron job "CSScron.dbload"

## Important Log Files



- **Gateway**
  - Gateway.log (used by Gateway and Gateway Proxy)
- **Ingest Subsystem**
  - IngestLocal.log (used by Ingest and Session Servers)
- **PDPS**
  - pdps\_event.log

520-TD-001-002 M&O 6-12

### Discussion Topics

Important Log Files -- data will be kept long-term on the AIT server (indefinitely)

- Gateway
  - Gateway.log (used by Gateway and Gateway Proxy)
- Ingest Subsystem
  - IngestLocal.log (used by Ingest and Session Servers)
- PDPS
  - pdps\_event.log